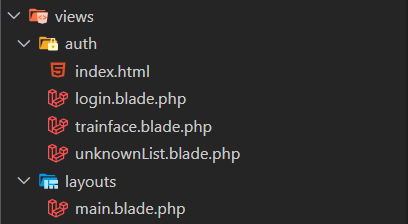
Chuyển chức năng nhận diện, huấn luyện từ Winform sang Laravel

# 1. View

## 1.1. Layout



**main.blade.php**

| <!DOCTYPE html> <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <meta name="csrf-token" content="{{ csrf\_token() }}">   <title>@yield('title', 'Default Title')</title>  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css"  integrity="sha512-1ycn6IcaQQ40/MKBW2W4Rhis/DbILU74C1vSrLJxCq57o941Ym01SwNsOMqvEBFlcgUa6xLiPY/NS5R+E6ztJQ=="  crossorigin="anonymous" referrerpolicy="no-referrer" />  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"  integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">   <!-- Yield cho các link CSS tùy chỉnh từ view -->  @yield('custom-css') </head>  <body>  <header>  <nav class="navbar navbar-expand-lg bg-body-tertiary">  <div class="container-fluid">  <a class="navbar-brand" href="#">LOGO</a>  <button class="navbar-toggler" type="button" data-bs-toggle="collapse"  data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"  aria-expanded="false" aria-label="Toggle navigation">  <span class="navbar-toggler-icon"></span>  </button>  <div class="collapse navbar-collapse" id="navbarSupportedContent">  <ul class="navbar-nav me-auto mb-2 mb-lg-0">  <li class="nav-item">  <a id="Home" class="nav-link active" aria-current="page" href="/">Home</a>  </li>  <li class="nav-item">  <a id="Login" class="nav-link" href="/login">Đăng nhập</a>  </li>  <li class="nav-item dropdown">  <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown"  aria-expanded="false">  Khác  </a>  <ul class="dropdown-menu">  <li><a class="dropdown-item" href="/train-face">Thêm dữ liệu khuôn mặt</a></li>  <li><a class="dropdown-item" href="/recognition-unknown-list">Khuôn mặt chưa được nhận  diện</a></li>  <li>  <hr class="dropdown-divider">  </li>  <li><a class="dropdown-item" href="#">Something else here</a></li>  </ul>  </li>  </ul>  <form class="d-flex" role="search">  <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">  <button class="btn btn-outline-success" type="submit">Search</button>  </form>  </div>  </div>  </nav>  </header>   <div class="container">  @yield('content')  </div>   <footer>  <!-- Footer content -->  </footer>   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"  integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r" crossorigin="anonymous">  </script>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"  integrity="sha384-0pUGZvbkm6XF6gxjEnlmuGrJXVbNuzT9qBBavbLwCsOGabYfZo0T0to5eqruptLy" crossorigin="anonymous">  </script>   @stack('scripts')  </body>  </html> |
| --- |

## 1.2. Trainface.blade.php

| @extends('layouts.main')  @section('title', 'Huấn luyện')  @section('custom-css')  <style>  .wrapper {  position: relative;  display: flex;  flex-direction: column;  align-items: center;  justify-content: center;  border-radius: 10px;  overflow: hidden !important;  }   #video {  transform: scaleX(-1);  background-color: #ccc;  border-radius: 10px;  overflow: hidden;  position: relative;  }   .controls {  text-align: center;  }   .messageInFrame {  position: absolute;  top: 0;  left: 0;  width: 100%;  height: 100%;  background-color: rgba(55, 55, 55, 0.5);  /\* Màu nền bán trong suốt \*/  display: flex;  align-items: center;  justify-content: center;  font-size: 1.5rem;  font-weight: bold;  color: black;  }   .fontLarge {  font-size: 6rem;  }   .img-thumbnail {  height: 100%;  width: auto;  object-fit: cover;  }  </style> @endsection  @section('content')  <div style="height: 100vh;" id="app" class="">  <div class="container">  <div class="row">  <div class="col-md-2">  <h4>Hướng dẫn huấn luyện khuôn mặt</h4>  <p>Bạn cần nhập username vào ô dữ liệu, sau đó nhấn huấn luyện. Hệ thống sẽ tự động chụp lấy 6 bức  ảnh khuôn mặt của bạn và tiến hành nhận diện</p>  <p>Mỗi một ảnh được chụp sẽ hiển thị bên cạnh khung hình video. Để dữ liệu khuôn mặt có tính xác  thực hơn bạn nên chụp các góc mặt như hướng dẫn sau đây:</p>  </div>  <div class="col-md-6">  <div id="TrainFace">  <div class="wrapper">  <video style="" class="" id="video" autoplay class="img-fluid rounded"></video>  <div class="messageInFrame text-white d-none">Thông điệp trên khung camera</div>  </div>  <canvas id="canvas" class="d-none"></canvas>  </div>  <div class="controls">  <div class="input-group my-1">  <input id="UsernameInput" type="text" class="form-control" placeholder="Username" required  aria-label="Username" aria-describedby="basic-addon2">  <div class="input-group-append">  <button id="TrainButton" class="btn btn-outline-secondary">Huấn luyện</button>  </div>  </div>  <div class="message text-danger d-none">Vui lòng nhập Username</div>  </div>  </div>  <div class="col-md-4">  <div class="row">  <div class="col-md-6 pb-2">  <img id="image1" class="img-thumbnail"  src="https://i.pinimg.com/564x/70/89/b2/7089b2489fbc7915346cd60120a25e0a.jpg"  alt="Ảnh không tồn tại!">  </div>  <div class="col-md-6 pb-2">  <img id="image2" class="img-thumbnail"  src="https://i.pinimg.com/564x/70/89/b2/7089b2489fbc7915346cd60120a25e0a.jpg"  alt="Ảnh không tồn tại!">  </div>  <div class="col-md-6 pb-2">  <img id="image3" class="img-thumbnail"  src="https://i.pinimg.com/564x/70/89/b2/7089b2489fbc7915346cd60120a25e0a.jpg"  alt="Ảnh không tồn tại!">  </div>  <div class="col-md-6 pb-2">  <img id="image4" class="img-thumbnail"  src="https://i.pinimg.com/564x/70/89/b2/7089b2489fbc7915346cd60120a25e0a.jpg"  alt="Ảnh không tồn tại!">  </div>  <div class="col-md-6 pb-2">  <img id="image5" class="img-thumbnail"  src="https://i.pinimg.com/564x/70/89/b2/7089b2489fbc7915346cd60120a25e0a.jpg"  alt="Ảnh không tồn tại!">  </div>  <div class="col-md-6 pb-2">  <img id="image6" class="img-thumbnail"  src="https://i.pinimg.com/564x/70/89/b2/7089b2489fbc7915346cd60120a25e0a.jpg"  alt="Ảnh không tồn tại!">  </div>  <div class="col-md-6 pb-2">  <img id="image7" class="img-thumbnail"  src="https://i.pinimg.com/564x/70/89/b2/7089b2489fbc7915346cd60120a25e0a.jpg"  alt="Ảnh không tồn tại!">  </div>  <div class="col-md-6 pb-2">  <img id="image8" class="img-thumbnail"  src="https://i.pinimg.com/564x/70/89/b2/7089b2489fbc7915346cd60120a25e0a.jpg"  alt="Ảnh không tồn tại!">  </div>  <div class="col-md-6 pb-2">  <img id="image9" class="img-thumbnail"  src="https://i.pinimg.com/564x/70/89/b2/7089b2489fbc7915346cd60120a25e0a.jpg"  alt="Ảnh không tồn tại!">  </div>  <div class="col-md-6 pb-2">  <img id="image10" class="img-thumbnail"  src="https://i.pinimg.com/564x/70/89/b2/7089b2489fbc7915346cd60120a25e0a.jpg"  alt="Ảnh không tồn tại!">  </div>  </div>  </div>   </div>  </div>  @endsection    @push('scripts')  <script src="{{ asset('Assets/js/trainface.js') }}"></script>  @endpush |
| --- |

**Assets/js/trainface.js**

| $(document).ready(async function () {  let stream;  async function startCamera() {  try {  stream = await navigator.mediaDevices.getUserMedia({ video: true });  const video = $('#video')[0];  video.srcObject = stream;  } catch (error) {  console.error('Lỗi khi truy cập camera:', error);  }  }  function stopCamera() {  if (stream) {  stream.getTracks().forEach(track => track.stop());  const video = $('#video')[0];  video.srcObject = null;  stream = null;  }  }  $('#TrainButton').click(async function () {  const username = $('#UsernameInput').val();  if (username === '') {  $('.message').removeClass('d-none').addClass('d-block');  return;  }  $('.message').removeClass('d-block').addClass('d-none');  try {  await displayMessage('Vui lòng nhìn vào màn hình', 3);  await takePhoto(1);  await delay(250);  await takePhoto(2);  await delay(250);  await takePhoto(3);  await delay(250);  await takePhoto(4);  await delay(250);  await takePhoto(5);  await $('#image1').attr('src', "ImageTrain/photo\_1.png");  await displayMessage('Vui lòng nhìn thẳng', 1);  await takePhoto(6);  await delay(250);  await takePhoto(7);  await delay(250);  await takePhoto(8);  await delay(250);  await takePhoto(9);  await delay(250);  await takePhoto(10);  await $('#image2').attr('src', "ImageTrain/photo\_6.png");  await displayMessage('Vui lòng hơi ngẩng mặt', 1);  await takePhoto(11);  await delay(250);  await takePhoto(12);  await delay(250);  await takePhoto(13);  await delay(250);  await takePhoto(14);  await delay(250);  await takePhoto(15);  await $('#image3').attr('src', "ImageTrain/photo\_11.png");  await displayMessage('Vui lòng hơi cúi mặt', 1);  await takePhoto(16);  await delay(250);  await takePhoto(17);  await delay(250)  await takePhoto(18);  await delay(250);  await takePhoto(19);  await delay(250);  await takePhoto(20);  await $('#image4').attr('src', "ImageTrain/photo\_16.png");  await displayMessage('Vui lòng nghiêng mặt sang trái', 1);  await takePhoto(21);  await delay(250);  await takePhoto(22);  await delay(250);  await takePhoto(23);  await delay(250);  await takePhoto(24);  await delay(250);  await takePhoto(25);  await $('#image5').attr('src', "ImageTrain/photo\_21.png");  await displayMessage('Vui lòng nghiêng mặt sang phải', 1);  await takePhoto(26);  await delay(250);  await takePhoto(27);  await delay(250);  await takePhoto(28);  await delay(250);  await takePhoto(29);  await delay(250);  await takePhoto(30);  await $('#image6').attr('src', "ImageTrain/photo\_26.png");  await displayMessage('Vui lòng ngẩng mặt và nghiêng sang trái', 1);  await takePhoto(31);  await delay(250);  await takePhoto(32);  await delay(250);  await takePhoto(33);  await delay(250);  await takePhoto(34);  await delay(250);  await takePhoto(35);  await $('#image7').attr('src', "ImageTrain/photo\_31.png");  await displayMessage('Vui lòng ngẩng mặt và nghiêng sang phải', 1);  await takePhoto(36);  await delay(250);  await takePhoto(37);  await delay(250);  await takePhoto(38);  await delay(250);  await takePhoto(39);  await delay(250);  await takePhoto(40);  await $('#image8').attr('src', "ImageTrain/photo\_36.png");  await displayMessage('Vui lòng cúi mặt và nghiêng sang trái', 1);  await takePhoto(41);  await delay(250);  await takePhoto(42);  await delay(250);  await takePhoto(43);  await delay(250);  await takePhoto(44);  await delay(250);  await takePhoto(45);  await $('#image9').attr('src', "ImageTrain/photo\_41.png");  await displayMessage('Vui lòng cúi mặt và nghiêng sang phải', 1);  await takePhoto(46);  await delay(250);  await takePhoto(47);  await delay(250);  await takePhoto(48);  await delay(250);  await takePhoto(49);  await delay(250);  await takePhoto(50);  await $('#image10').attr('src', "ImageTrain/photo\_46.png");  await displayMessage('Đang xử lý', 'Đang xử lý');  $('.messageInFrame').text("Đang xử lý...").removeClass('d-none').addClass('d-flex');  await $.ajax({  url: '/photo-train',  method: 'POST',  headers: {  'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content'),  },  data: { username },  success: function (response) {  console.log(response.commands);  $('.messageInFrame').removeClass('d-flex').addClass('d-none');  },  error: function (error) {  console.error('Lỗi:', error);  $('.messageInFrame').text('Đã xảy ra lỗi. Vui lòng thử lại sau.').removeClass('d-none').addClass('d-flex');  }  });  $('.messageInFrame').removeClass('d-flex').addClass('d-none');  } catch (error) {  console.error('Lỗi:', error);  }  });  async function delay(ms) {  return new Promise(resolve => setTimeout(resolve, ms));  }  async function takePhoto(count) {  const video = $('#video')[0];  const canvas = $('#canvas')[0];  const context = canvas.getContext('2d');  canvas.width = video.videoWidth;  canvas.height = video.videoHeight;  context.translate(canvas.width, 0);  context.scale(-1, 1);  context.drawImage(video, 0, 0, canvas.width, canvas.height);  const imageBase64 = canvas.toDataURL('image/png');  try {  const response = await $.ajax({  url: '/save-photo-train',  method: 'POST',  headers: {  'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content'),  'Content-Type': 'application/json',  },  data: JSON.stringify({ imageBase64, count }),  });  console.log('Ảnh đã được lưu:', response.fileUrl);  } catch (error) {  console.error('Lỗi:', error);  }  }  async function displayMessage(message, count) {  $('.messageInFrame').removeClass('d-none').text(message);  if (!isNaN(count)) {  $('.messageInFrame').text(message);  await new Promise(resolve => setTimeout(resolve, 1000));  for (let i = count; i >= 0; i--) {  $('.messageInFrame').text(i).addClass('fontLarge');  await new Promise(resolve => setTimeout(resolve, 1000));  }  }  $('.messageInFrame').text('').removeClass('fontLarge');  $('.messageInFrame').addClass('d-none');  }  // Bắt đầu camera khi tài liệu sẵn sàng  await startCamera();  // Dừng camera khi rời khỏi trang  $(window).on('unload', function () {  stopCamera();  });  // Xóa ảnh khi tải lại trang  $(window).on('load', function () {  stopCamera();  $.ajax({  url: '/delete-images',  method: 'GET',  success: function (response) {  console.log(response.message);  },  error: function (xhr, status, error) {  console.error('Lỗi:', error);  }  });  });  }); |
| --- |

## 1.3. Login.blade.php

| @extends('layouts.main')  @section('title', 'Đăng nhập')  @section('custom-css')  <link rel="stylesheet" href="{{ asset('Assets/css/login.css') }}">  @endsection  @section('content')  <div class="container mt-5">  <div class="row justify-content-center">  <div class="col-md-6">  <div class="card">  <div class="card-header">Đăng nhập</div>  <div class="card-body">  <form method="POST" action="{{ route('login.submit') }}">  @csrf  <div class="form-group">  <label for="username">Tên đăng nhập:</label>  <input type="text" class="form-control" id="username" name="username"  value="{{ old('username') }}" required autofocus>  </div>  <div class="form-group">  <label for="password">Mật khẩu:</label>  <input type="password" class="form-control" id="password" name="password" required>  </div>  <button id="ButtonLogin" type="submit" class="btn btn-primary">Đăng nhập</button>  @error('loginError')  <div class="text-danger mt-2">{{ $message }}</div>  @enderror  </form>  </div>  <div class="card-footer">  <button type="button" class="btn btn-success" id="btnFaceLogin">Đăng nhập bằng khuôn  mặt</button>  </div>  </div>  </div>  <div class="col-md-6">  <div id="Recognize" class="recognizeface w-100 h-100 d-none">  <div class="wrapper">  <video class="w-100" id="video" autoplay class="img-fluid rounded"></video>  <div id="loadingIndicator" class="d-none" style="text-align: center;">  <img src="./assets/images/loading.gif" alt="Loading..." />  </div>  </div>  <div class="controls">  <button id="recognizeButton" class="btn btn-success">Nhận dạng</button>  <h1 id="personName" class="d-none"></h1>  </div>  <canvas id="canvas" class="d-none"></canvas>  </div>  </div>  </div>  @if (session('message') && session('type'))  <div class="toast-container rounded position-fixed bottom-0 end-0 p-3">  <div id="liveToast" class="toast" role="alert" aria-live="assertive" aria-atomic="true">  <div class="toast-body bg-{{ session('type') }} d-flex align-items-center justify-content-between">  <div class="d-flex justify-content-center align-items-center gap-2">  @if (session('type') == 'success')  <i class="fas fa-check-circle text-light fs-5"></i>  @elseif(session('type') == 'danger' || session('type') == 'warning')  <i class="fas fa-times-circle text-light fs-5"></i>  @elseif(session('type') == 'info' || session('type') == 'secondary')  <i class="fas fa-info-circle text-light fs-5"></i>  @endif  <h6 class="h6 text-white m-0">{{ session('message') }}</h6>  </div>  <button type="button" class="btn-close btn-close-white" data-bs-dismiss="toast"  aria-label="Close"></button>  </div>  </div>  </div>  @endif  </div>  @endsection  @push('scripts')  <script>  $(document).ready(function() {  let stream;  // Hàm bắt đầu camera  async function startCamera() {  try {  stream = await navigator.mediaDevices.getUserMedia({  video: true  });  const video = $('#video')[0];  video.srcObject = stream;  } catch (error) {  console.error('Lỗi khi truy cập camera:', error);  }  }  // Hàm dừng camera  function stopCamera() {  if (stream) {  stream.getTracks().forEach(track => track.stop());  const video = $('#video')[0];  video.srcObject = null;  stream = null;  }  }  // Xử lý sự kiện click vào nút "Đăng nhập bằng khuôn mặt"  $('#btnFaceLogin').click(function() {  $('#Recognize').toggle(); // Hiển thị hoặc ẩn phần nhận dạng khuôn mặt  if (stream) {  stopCamera();  $('#Recognize').removeClass('d-block').addClass('d-none');  } else {  startCamera();  $('#Recognize').removeClass('d-none').addClass('d-block');  setTimeout(function() {  takephoto();  }, 1000);  }  });  // Xử lý sự kiện click vào nút "Nhận dạng"  $('#recognizeButton').click(function() {  takephoto();  });  function takephoto() {  // Lấy video và canvas elements  const video = $('#video')[0];  const canvas = $('#canvas')[0];  const context = canvas.getContext('2d');  // Đặt kích thước canvas bằng với kích thước video  canvas.width = video.videoWidth;  canvas.height = video.videoHeight;  // Đảo ngược hình ảnh nếu cần thiết  context.translate(canvas.width, 0);  context.scale(-1, 1);  // Vẽ khung hình hiện tại của video lên canvas  context.drawImage(video, 0, 0, canvas.width, canvas.height);  // Lấy dữ liệu ảnh từ canvas dưới dạng base64  const imageBase64 = canvas.toDataURL('image/png');  $('#loadingIndicator').removeClass('d-none').addClass('d-block');  $('#personName').text("");  // Gửi dữ liệu ảnh base64 đến máy chủ qua AJAX  $.ajax({  url: '/save-photo',  method: 'POST',  headers: {  'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content'),  'Content-Type': 'application/json'  },  data: JSON.stringify({  imageBase64: imageBase64  }),  success: function(response) {  console.log('Ảnh đã được lưu:', response.filepath);  recognizeFace(response.filepath);  },  error: function(xhr, status, error) {  console.error('Lỗi:', error);  }  });  }  function recognizeFace(imagePath) {  $.ajax({  url: '/recognize-face',  method: 'POST',  headers: {  'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content'),  'Content-Type': 'application/json'  },  data: JSON.stringify({  imagePath: imagePath  }),  success: function(response) {  console.log('Tên người được nhận dạng:', response.recognizedName);  $('#personName').text(response.recognizedName).removeClass('d-none').addClass(  'd-block');  if (response.recognizedName !== 'Unknown' && response.recognizedName !==  'Không có khuôn mặt được tìm thấy!' && response.recognizeFace !=  'Phát hiện 2 khuôn mặt, vui lòng thử lại!') {  window.location.href = '/';  } else {  alert('Tên đăng nhập hoặc mật khẩu không chính xác!');  }  },  error: function(xhr, status, error) {  console.error('Lỗi:', error);  },  complete: function() {  $('#loadingIndicator').removeClass('d-block').addClass('d-none');  }  });  }  });  </script>  @endpush |
| --- |

## 1.4. UnknowList.blade.php

| @extends('layouts.main')  @section('title', 'Danh sách khuôn mặt chưa biết')  @section('content')  <div class="container">  <h2>Danh sách khuôn mặt chưa biết</h2>  <table class="table">  <thead class="thead-dark">  <tr>  <th scope="col">STT</th>  <th scope="col">Hình ảnh</th>  <th scope="col">Ngày nhận diện</th>  <th scope="col">Huấn luyện</th>  <th scope="col">Xóa</th>  </tr>  </thead>  <tbody>  @foreach ($unknownRecognitions as $recognition)  <tr>  <td>{{ $recognition['id'] }}</td>  <td><img src="{{ $recognition['image'] }}" class="img-thumbnail" alt="Image"  style="max-width: 100px;"></td>  <td>{{ $recognition['date'] }}</td>  <td>  <button type="button" class="btn btn-primary btn-train" data-file="{{ $recognition['image'] }}"  data-bs-toggle="modal" data-bs-target="#trainModal">  Huấn luyện  </button>  </td>  <td>  <button class="btn btn-danger btn-delete" data-file="{{ $recognition['image'] }}">  <i class="fas fa-trash"></i>  </button>  </td>  </tr>  @endforeach  </tbody>  </table>  </div>  <!-- Thêm modal này vào sau phần table -->  <div class="modal fade" id="trainModal" tabindex="-1" aria-labelledby="trainModalLabel" aria-hidden="true">  <div class="modal-dialog">  <div class="modal-content">  <div class="modal-header">  <h5 class="modal-title" id="trainModalLabel">Huấn luyện khuôn mặt</h5>  <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>  </div>  <div class="modal-body">  <form id="trainForm">  <div class="mb-3">  <label for="faceUserName" class="form-label">Username</label>  <input type="text" class="form-control" id="faceUserName" name="faceUserName" required>  </div>  <input type="hidden" id="recognitionId" name="recognitionId">  <button type="submit" id="submit" class="btn btn-primary">Lưu</button>  </form>  </div>  </div>  </div>  </div>  @endsection  @push('scripts')  <script>  $(document).ready(function() {  var filePath;  $('.btn-delete').click(function(e) {  e.preventDefault();  filePath = $(this).data('file');  var token = "{{ csrf\_token() }}";  $.ajax({  url: "{{ route('delete.image') }}",  type: 'POST',  dataType: 'json',  data: {  \_token: token,  filePath: filePath // Chắc chắn rằng đường dẫn được gửi đi đúng định dạng  },  success: function(response) {  if (response.success) {  // Tải lại danh sách sau khi xóa thành công  location.reload();  } else {  console.error(response.error);  }  },  error: function(xhr, status, error) {  console.error(xhr.responseText);  // Xử lý lỗi  }  });  });  // Xử lý sự kiện huấn luyện  $('.btn-train').click(function() {  var recognitionId = $(this).closest('tr').find('td:first').text();  $('#recognitionId').val(recognitionId);  filePath = $(this).data('file');  });  // Xử lý sự kiện huấn luyện  $('#trainForm').submit(function(e) {  e.preventDefault();  const username = $('#faceUserName').val();  if (username !== '') {  console.log(filePath);  console.log(username);  $.ajax({  url: '/photo-train-image', // Đường dẫn xử lý huấn luyện  method: 'POST',  headers: {  'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content'),  },  data: {  username: username,  filePath: filePath  },  success: function(response) {  console.log(response);  $('#trainModal').modal('hide');  location.reload();  },  error: function(error) {  console.error('Lỗi:', error);  // Xử lý lỗi  }  });  }  });  });  </script>  @endpush |
| --- |

# 2. Route

| <?php use App\Http\Controllers\AuthController; use App\Http\Controllers\TrainController; use Illuminate\Support\Facades\Route;  Route::get('/', function () {  return view('welcome'); });  Route::get('/login', [AuthController::class, 'showLoginForm'])->name('login'); Route::post('/login', [AuthController::class, 'login'])->name('login.submit');  // Huấn luyện khuôn mặt Route::post('/photo-train', [TrainController::class, 'TrainAllFace']); Route::get('/train-face', function () {  return view('auth/trainface'); }); Route::post('/save-photo-train', [TrainController::class, 'savePhoto']); Route::post('/photo-train-image', [TrainController::class, 'TrainFace']);  // Nhận diện khuôn mặt Route::post('/save-photo', [AuthController::class, 'savePhoto']); Route::post('/recognize-face', [AuthController::class, 'recognizeFace']); Route::get('/delete-images', [TrainController::class, 'deleteImages'])->name('delete.images');  // Khuôn mặt chưa biết Route::get('/recognition-unknown-list', [TrainController::class, 'UnknownList']); Route::post('/delete-image', [TrainController::class, 'deleteImage'])->name('delete.image'); |
| --- |

# 3. Controller

chạy **php artisan storage:link**

## 3.1. AuthController

| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  use Illuminate\Support\Facades\Storage;  class AuthController extends Controller  {  // Hiển thị form đăng nhập  public function showLoginForm()  {  return view('auth.login');  }  // Xử lý đăng nhập  public function login(Request $request)  {  return redirect('/welcome'); // Thay thế '/dashboard' bằng route bạn muốn chuyển hướng đến khi đăng nhập thành công  }  public function savePhoto(Request $request)  {  if ($request->ajax()) {  if ($request->has('imageBase64')) {  $imageData = $request->input('imageBase64');  // Chuẩn bị dữ liệu ảnh để giải mã  $imageData = str\_replace('data:image/png;base64,', '', $imageData); // Loại bỏ phần header của base64  $imageData = str\_replace(' ', '+', $imageData); // Thay thế các khoảng trắng  // Giải mã dữ liệu base64 thành dữ liệu nhị phân của ảnh  $imageBinary = base64\_decode($imageData);  // Đường dẫn tới thư mục trong storage/app  $uploadPath = 'ImageRecognize/';  // Tạo thư mục nếu chưa tồn tại  if (!Storage::exists($uploadPath)) {  Storage::makeDirectory($uploadPath, 0777, true, true);  }  // Tạo tên file duy nhất  $filename = 'photo\_' . date('Y-m-d-H-i-s') . '.png';  // Lưu ảnh vào thư mục trong storage  Storage::put($uploadPath . $filename, $imageBinary);  // Lấy đường dẫn tuyệt đối của file đã lưu  $filePath = Storage::path($uploadPath . $filename);  // Trả về đường dẫn tuyệt đối  return response()->json(['filepath' => realpath($filePath)]);  } else {  return response()->json(['error' => 'Không có dữ liệu ảnh được gửi lên.'], 400);  }  }  }  public function recognizeFace(Request $request)  {  if ($request->ajax()) {  if ($request->has('imagePath')) {  $imagePath = $request->input('imagePath');  // Kiểm tra và lấy đường dẫn chính xác của các file và script Python  $pythonScriptPath = storage\_path('app/python/FaceRecognition.py');  $encodingPath = storage\_path('app/models/encodings.txt');  $outputPath = storage\_path('app/data/output.txt');  $command = escapeshellcmd("python $pythonScriptPath recognize\_faces $imagePath $encodingPath $outputPath");  // Thực hiện lệnh và lưu kết quả  exec($command, $output[], $returnVars[]);  // Đọc kết quả từ file output  if (file\_exists($outputPath)) {  $recognizedName = trim(file\_get\_contents($outputPath));  if ($recognizedName === 'Unknown') {  // Nếu nhận dạng là "Unknown", di chuyển ảnh vào thư mục public/Storage/ImageUnknown  $newImagePath = public\_path('Storage/ImageUnknown') . '/' . basename($imagePath);  copy($imagePath, $newImagePath);  }    return response()->json([  'command' => $command,  'recognizedName' => $recognizedName,  'imagePath' => $imagePath,  'encodingPath' => $encodingPath,  'outputPath' => $outputPath,  'pythonScriptPath' => $pythonScriptPath,  ]);  } else {  return response()->json(['error' => 'Không tìm thấy file kết quả.'], 404);  }  } else {  return response()->json(['error' => 'Không có dữ liệu imagePath được gửi lên.'], 400);  }  }  }  } |
| --- |

## 3.2. TrainController

| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  use Illuminate\Support\Facades\Storage;  class AuthController extends Controller  {  // Hiển thị form đăng nhập  public function showLoginForm()  {  return view('auth.login');  }  // Xử lý đăng nhập  public function login(Request $request)  {  return redirect('/welcome'); // Thay thế '/dashboard' bằng route bạn muốn chuyển hướng đến khi đăng nhập thành công  }  public function savePhoto(Request $request)  {  if ($request->ajax()) {  if ($request->has('imageBase64')) {  $imageData = $request->input('imageBase64');  // Chuẩn bị dữ liệu ảnh để giải mã  $imageData = str\_replace('data:image/png;base64,', '', $imageData); // Loại bỏ phần header của base64  $imageData = str\_replace(' ', '+', $imageData); // Thay thế các khoảng trắng  // Giải mã dữ liệu base64 thành dữ liệu nhị phân của ảnh  $imageBinary = base64\_decode($imageData);  // Đường dẫn tới thư mục trong storage/app  $uploadPath = 'ImageRecognize/';  // Tạo thư mục nếu chưa tồn tại  if (!Storage::exists($uploadPath)) {  Storage::makeDirectory($uploadPath, 0777, true, true);  }  // Tạo tên file duy nhất  $filename = 'photo\_' . date('Y-m-d-H-i-s') . '.png';  // Lưu ảnh vào thư mục trong storage  Storage::put($uploadPath . $filename, $imageBinary);  // Lấy đường dẫn tuyệt đối của file đã lưu  $filePath = Storage::path($uploadPath . $filename);  // Trả về đường dẫn tuyệt đối  return response()->json(['filepath' => realpath($filePath)]);  } else {  return response()->json(['error' => 'Không có dữ liệu ảnh được gửi lên.'], 400);  }  }  }  public function recognizeFace(Request $request)  {  if ($request->ajax()) {  if ($request->has('imagePath')) {  $imagePath = $request->input('imagePath');  // Kiểm tra và lấy đường dẫn chính xác của các file và script Python  $pythonScriptPath = storage\_path('app/python/FaceRecognition.py');  $encodingPath = storage\_path('app/models/encodings.txt');  $outputPath = storage\_path('app/data/output.txt');  $command = escapeshellcmd("python $pythonScriptPath recognize\_faces $imagePath $encodingPath $outputPath");  // Thực hiện lệnh và lưu kết quả  exec($command, $output[], $returnVars[]);  // Đọc kết quả từ file output  if (file\_exists($outputPath)) {  $recognizedName = trim(file\_get\_contents($outputPath));  if ($recognizedName === 'Unknown') {  // Nếu nhận dạng là "Unknown", di chuyển ảnh vào thư mục public/Storage/ImageUnknown  $newImagePath = public\_path('Storage/ImageUnknown') . '/' . basename($imagePath);  copy($imagePath, $newImagePath);  }    return response()->json([  'command' => $command,  'recognizedName' => $recognizedName,  'imagePath' => $imagePath,  'encodingPath' => $encodingPath,  'outputPath' => $outputPath,  'pythonScriptPath' => $pythonScriptPath,  ]);  } else {  return response()->json(['error' => 'Không tìm thấy file kết quả.'], 404);  }  } else {  return response()->json(['error' => 'Không có dữ liệu imagePath được gửi lên.'], 400);  }  }  }  } |
| --- |